

**REMARKS**

Claims 1-5, 8-9, and 11-12 have been amended, no claims have been cancelled and claims 14-20 have been added. Thus, claims 1-20 are pending. No new matter has been added.

**Attorney Docket Number**

Applicant draws the Examiner's attention to the Attorney Docket Number of this Application which should now be listed as MWS-074.

**Objection to the Drawings**

Formal drawings have been prepared and are submitted herewith.

**Claim Objection**

Claim 9 was objected to for an improper dependency. An appropriate correction has been made.

**Claim Rejections Pursuant to 35 U.S.C. §103(a)**

Claims 1-3, 5, 9-13 were rejected by the Examiner pursuant to 35 U.S.C. §103(a) as being unpatentable over McKaskle et al (United States Patent Number 5,481, 741 hereafter "McKaskle") in view of Uczekaj et al (United States Patent Number 5,708, 825, hereafter Uczekaj"). In view of the above amendments and the below remarks, those rejections are respectfully traversed.

**Summary of the Claimed Invention**

Applicant claims a sub-classing mechanism used in graphical block diagram models. A graphical class corresponds to a graphical subsystem block made up of a plurality of interconnected graphical blocks. A graphical class instance of the subsystem block is instantiated in a user's block diagram. The graphical class instance includes a link to the parent graphical class. The graphical class instance includes an interface enabling the user to make

changes to the parameter values of the instantiated block and/or its constituent block components. The changes are applied to the graphical class instance which is linked to the parent class to produce a subclass instance that inherits an updated structure from the library graphical class and applies the changed parameter. In this manner, a user may independently change subsystem block parameters while receiving the benefit of any changes/updates to the graphical class held in the class library.

#### Summary of Claim Amendments

Independent claims 1 and 11 have been amended to better clarify that the process of changing a parameter for one of the subsystem blocks in the subsystem results in the updating of the structure inherited by the subclass instance to account for any recent changes to the graphical class in the library. Claim 8 has been amended to indicate that the merging process may propagate the changes from the subclass to the graphical library class or re-instantiate the graphical library class as the subclass. New claims 14-20 are medium claims corresponding to method claims 2-8.

#### Summary of McKaskle

McKaskle discusses a system and method of programmatically accessing various parameters of a control or indicator in a front panel linked to a data flow diagram. Changes may be made by the user affecting the appearance or output of controls and indicators in the front panel. The changes may be made during the execution of the block diagram associated with the front panel. McKaskle discusses a virtual instrument which includes an interconnected block diagram editor, execution subsystem, icon editor and front panel editor. The front panel permits interactive use of the virtual instrument by a user as it includes graphical representations for input (control) and output (indicator) values for a process performed by the linked block diagram. The virtual instrument may be represented via an icon as a sub-unit in the block diagram of another virtual instrument. Parameter changes are made via attribute node objects representing the controls/indicators in the front panel.

Summary of Uczekaj

Uczekaj discusses a programmatic code generation system. A graphical user interface is used to enter object information and associated control information. The system automatically generates application shell code according to the operating system environment in which the system is running and the user entered information. The invention thus avoids hand coding application code for each object defined in the application. Uczekaj does not disclose a block diagram environment generally or block diagram subsystems specifically.

Argument

The combination of Mckaskle and Uczekaj fails to disclose teach or suggest all of the elements of Applicants claims. The claimed invention is directed towards the problem of allowing customization of subsystem block parameters of an instantiated subsystem block object while also maintaining a link to a parent class in order to maintain the latest updated structure indicated by the parent class. The Examiner's attention is directed towards the Background of the present Application wherein it states:

“Prior block diagram modeling and simulation software packages...allow only values of top-level parameters of an instance to be changed. Typically, to modify an individual block within a graphical subsystem block in a library (and therefore model behavior of the subsystem block), a user has to make a local copy and apply any changes to that local copy. This approach is problematic, however, in that the local copy does not inherit any changes or improvements made to the original class. As a result, the local copy (in essence, a new class) quickly diverges from the original class and inclusion of the changes to the original class requires that the new class be updated manually. Another approach would be to specify all parameters of all of a subsystem block’s components at the top level. This can however, get extremely complicated for non-trivial classes and prove to be unusable.(See page 2 lines 1-11)

Applicant's claimed invention addresses this problem by instantiating an instance of the graphical class corresponding to a subsystem block found in the class library and a user-supplied parameter. A user interface allows a user to change a parameter for the subsystem component. The change in parameter causes a new subclass instance to be instantiated which reflects the changed parameter value and any updates to the structure of the linked graphical class. Neither McKaskle or Uczekaj disclose, teach or suggest this functionality.

Specifically, amended claim 1 (upon which claims 1-9 are dependent) and the corresponding medium claim 11, upon which claims 14-20 are dependent have been amended to clarify that the new graphical class instance is constructed from both the user-submitted changed parameter for the subsystem block component as well as inheriting the updated structure from the linked library graphical class. Neither McKaskle or Uczekaj reveals this feature. The Examiner admits that McKaskle does not disclose the limitation of constructing from the graphical class instance and the change a graphical subclass instance that inherits structure from the graphical class( see page 4 first paragraph of Office Action). The Examiner instead relies on Uczekaj as disclosing the limitation. For the reasons set forth below, Applicants respectfully disagree.

The Examiner cites language in column 5 of Uczekaj, lines 1-4 and 10-13, which discusses generally classes, subclasses and inheritance as disclosing the claimed feature of constructing the graphical subclass instance that inherits the updated structure of the parent graphical class and includes the changed parameter. This statement is unsupported by the reference. First, Uczekaj is not discussing a graphical programming/block diagram environment but rather the use of a graphical user interface to accept object and control information in order to subsequently programmatically generate application shell code. Also, Uczekaj does not discuss instantiation of subsystem or any other objects in response to a user changed parameter. Additionally, Uczekaj does not discuss subclassing a block diagram component of a block diagram subsystem. In fact, other than the general discussion of subclassing, Uczekaj is completely off point. Applicant is not claiming the general concept of subclassing but rather the use of subclassing in a graphical block diagram environment involving subsystems(i.e. the subclassing of a previously instantiated instance of a graphical block corresponding to a subsystem which uses both a different parameter supplied by a user and updated structure supplied by the library class, the fourth step of claims 1 and 11). The creation of the subclass instance which maintains symmetry with the graphical class in the library at the same time as it implements new user parameters is simply not discussed in either reference. It should also be pointed out that since there is nothing related to the block diagram environment of Applicant's invention in Uczekaj, there is no motivation to combine the general concepts of Uczekaj with McKaskle which is not directed towards solving the same problem as Applicant's invention.

Furthermore, neither McKaskle or Uczekaj discusses subsystem blocks in the same manner as claimed by the Applicant. As noted above, Uczekaj is not discussing a block diagram environment but rather has been cited by the Examiner for its general discussion of inheritance and subclassing. Applicant respectfully disagrees with the Examiner's contention that the section of McKaskle cited at col. 38, lines 50-55 reveals the subsystem blocks of Applicant's claims. The cited section is discussing generally the flow of data through the block diagram as a result of the code associated with the block diagram components being executed. The subsystem block of Applicant's claims corresponds to a plurality of blocks and is represented by a graphical class in the library. McKaskle discusses a block diagram tied to a front panel with inputs and output controls on the front panel corresponding to diagram components. The block diagram in McKaskle may have an additional sub-virtual instrument (with another front panel) referenced by an icon. McKaskle also discusses using attribute nodes to change parameters for front panel controls associated with diagram components. McKaskle does not discuss instantiating or subclassing subsystem blocks or components of subsystem blocks as claimed in Applicant's invention. The combination of references thus fails to disclose, teach or suggest the "construction of a graphical class instance of a graphical class that corresponds to the graphical subsystem block"(the second step of claims 1 and 11). Similarly, the combination of references fails to disclose, teach or suggest the changing of parameters in a subsystem block (third step of claims 1 and 11). Accordingly, since the combination of references fails to disclose all of the elements of independent claim 1 (and claims 2-9 which are dependent thereon) and claim 11 (and new claims 14-20 which are dependent thereon), Applicant requests the rejections directed to those claims be withdrawn and the claims allowed.

Similarly, independent claims 10 (method), and 12 and 13 (system) all include forms of the subclass limitation discussed above which involve the inheriting of changes to the graphical class and the use of the new parameter. For the same reasons discussed above, the combination of McKaskle and Uczekaj fail to disclose, teach or suggest these limitations. Accordingly, since the combination of references fails to disclose all of the elements of independent claims 10, 12 and 13, Applicant requests the rejections directed to those claims be withdrawn and the claims allowed.

As the dependent claims include all of the limitations of the underlying independent claims, it is respectfully suggested that they are now all in condition for allowance. In addition however, regarding claim 2, Applicant notes that the user interface in McKaskle is for a control on the front panel not for the block diagram per se and therefore does not disclose, teach or suggest the claimed limitation of a user interface for accepting a parameter for a subsystem block. The limitation of claim 3 involving the storing of subclass data in the data structure is not disclosed, taught or suggested by Uczekaj which only discusses subclassing in general terms and does not discuss the storing of subclass data. Uczekaj also does not disclose, teach or suggest the merging limitation of the amended claim 5 which includes the propagating of the subclass data to the graphical class and the re-instantiation of the graphical class in place of the subclass instance. The general high-level discussion of subclassing as a concept contained in Uczekaj is not a valid basis for maintaining these rejections.

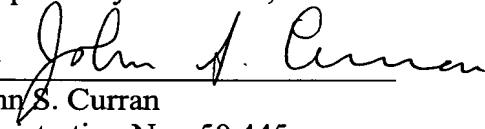
**CONCLUSION**

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 12-0080, under Order No. MWS-074 from which the undersigned is authorized to draw.

Dated: May 10, 2004

Respectfully submitted,

By   
John S. Curran  
Registration No.: 50,445  
LAHIVE & COCKFIELD, LLP  
28 State Street  
Boston, Massachusetts 02109  
(617) 227-7400  
(617) 742-4214 (Fax)  
Attorney/Agent For Applicant